DISTRIBUTED SPARSIFIED GRAPH FILTERS FOR DENOISING AND DIFFUSION TASKS

Elvin Isufi and Geert Leus

Circuits and Systems Group, Delft University of Technology, The Netherlands.
e-mails: {e.isufi-1, g.j.t.leus}@tudelft.nl

ABSTRACT

Generally in distributed signal processing, and specifically in distributed graph filters, reducing the communication and computational complexity plays a key role in the network lifetime. In this work we present a novel algorithm to sparsify the graph filtering operation in a random way, where each node decides locally with a certain probability with which of its neighbors to communicate. We show that, if the filter coefficients are changed accordingly, the first and second order moment of the stochastic output are identical to the deterministic filter output and bounded, respectively. We apply our idea on the tasks of signal denoising and diffusion. Numerical results show that the distributed implementation costs of the filter can be reduced up to 95% with a variance of 10^{-3} from the deterministic output.

Index Terms— graph signal processing, graph filters, graph signal denoising, diffusion graph signals, graph sparsification.

1. INTRODUCTION

Signals collected by a sensor network, or present in social networks and other irregular domains are often characterized by complex relationships captured by a graph [1]. Recent advances in the field of signal processing on graphs allow analyzing these signals not only in the vertex, but also in the graph spectral domain [2, 3]. Graph spectral domain processing is now made possible through the use of the graph Fourier transform and graph filters [2, 3]. The latter are direct analogous of temporal filters now operating on the graph frequency content of the signal [4]. Among others, graph filters are useful to clean noisy collected data (like sensor measurments) [2, 5] and also to analyze the diffusion of graph signals over the network (temperature diffusion, gossiping in a social network) [2, 6].

The distributed implementation of graph filters [7, 8] rises with the necessity to alleviate the computational cost of processing large amounts of data, or when, in a sensor network, due to physical limitation (communication range and the necessity to cover large areas) the sensors cannot deliver the data to a fusion center for processing. However, even in a distributed fashion, the communication and computational costs may be too expensive for cheap sensors.

In this work, we present a novel approach to reduce the distributed filtering cost by filtering the signal, rather than on the original graph, on its sparsified version. Specifically, in each iteration round of the filtering process each node will locally decide to exchange information randomly with its neighbors. The proposed approach can be seen as gossiping [9], while performing the graph filter, but in the same time also as sparsified filtering [10]. We will refer to this approach as *stochastic sparsification*. As a benefit the distributed costs are alleviated, but at the same time the filter output is now stochastic. We characterize the first and second order moment of the filter output. We show that, with a proper change of the filter coefficients, the expected filter output obtained in the *stochastic sparsified* graph is the same as that of the deterministic graph and that the average variance among all nodes is upper bounded.

In this work we will specifically focus on two main graph filtering tasks : Tikhonov denoising and distributed diffusion. However, note that the general idea can be applied to any graph filter and desired frequency response. Numerical results show that up to 95% of the communication and computational complexity can be saved with very little difference from the deterministic filter output.

2. BACKGROUND

In this section we recall some basics on signal processing on graphs, distributed graph filters, Tikhonov denoising and graph signal diffusion. We also show that for the aforementioned tasks a distributed finite impulse response (FIR) graph filter of any order can be alternatively implemented by an autoregressive moving average graph filter of first order (ARMA$_1$).

Signal processing on graphs. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ the set of N vertices and $\mathcal{E}$ the set of M edges. The local structure of $\mathcal{G}$ is represented by the *graph shift* operator $\boldsymbol{S}$, an $N \times N$ symmetric matrix with $S_{i,j} \neq 0$ if there exists an edge between the nodes i and j. Common choices[1] for $\boldsymbol{S}$ are the adjacency matrix of the graph [3], the graph discrete $\boldsymbol{L}_{\mathrm{d}}$ and normalized $\boldsymbol{L}_{\mathrm{n}}$ Laplacian [2] or their translated versions [**?**]. Being a symmetric matrix, $\boldsymbol{S}$ always enjoys an eigendecomposition $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{H}}$ with eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ and eigenvalues $\boldsymbol{\Lambda} = \mathrm{diag}[\lambda_1, \ldots, \lambda_N]$, which carry the notion of frequency in the graph setting [2, 3]. More formally, the eigenvalues $\{\lambda_n\}_{n=1}^{N}$ indicate the graph frequencies and the eigenvector matrix $\boldsymbol{U}$ is used as the graph Fourier expansion basis.

A *graph signal* is defined as the $N \times 1$ vector $\boldsymbol{x}$ with i-th entry $x_i \in \mathbb{C}$ living on the i-th node of $\mathcal{G}$. The graph Fourier transform $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ and its inverse are respectively calculated as $\hat{\boldsymbol{x}} = \boldsymbol{U}^{\mathsf{H}}\boldsymbol{x}$ and $\boldsymbol{x} = \boldsymbol{U}\hat{\boldsymbol{x}}$.

Distributed FIR graph filters. From [3], an FIR graph filter of order K (FIR$_K$) can be expressed as a K-th order polynomial of $\boldsymbol{S}$. The output signal and the graph frequency response of the filter are respectively given by

$$\boldsymbol{y} = \sum_{k=0}^{K} \phi_k \boldsymbol{S}^k \boldsymbol{x} \quad \text{and} \quad h(\lambda_n) = \sum_{k=0}^{K} \phi_k \lambda_n^k, \tag{1}$$

where ϕ_k indicate the filter coefficients. Due to the locality of the shift operator $\boldsymbol{S}$ and since $\boldsymbol{S}^K\boldsymbol{x}$ can be expressed as $\boldsymbol{S}(\boldsymbol{S}^{K-1}\boldsymbol{x})$, each node can compute the K-th term from the values of the $(K-1)$-th terms in its neighborhood [7]. This leads to a distributed communication and computational cost of $O(MK)$.

Distributed ARMA$_1$ graph filters. An ARMA$_1$ graph filter [8] can be implemented in a distributed fashion as

$$\boldsymbol{y}_t = \psi \boldsymbol{S} \boldsymbol{y}_{t-1} + \varphi \boldsymbol{x}, \tag{2}$$

where $\boldsymbol{x}$ is the graph signal to be filtered, $\boldsymbol{y}_t$ is the filter output at time t with arbitrary initial condition $\boldsymbol{y}_0$ and where ψ and φ indicate

[1] For this work, we only require the shift operator to have an upper bounded spectral norm, i.e., $\|\boldsymbol{S}\| \leq \varrho$ for some $\varrho > 0$.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

the filter coefficients. From [8], the graph frequency response of the ARMA$_1$ filter is

$$h(\lambda_n) = \frac{\varphi}{1-\psi\lambda_n} \quad \text{subject to} \quad |\psi|< \varrho, \tag{3}$$

with ϱ the upper bound on the spectral norm of $\boldsymbol{S}$. As shown in [8], recursion (2) will attain the frequency response (3) theoretically at infinity, yet in practice characterized by a linear convergence. The $per-iteration$ communication cost of the ARMA$_1$ filter is $O(M)$.

Tikhonov denoising. Graph signal denoising under a smoothness prior [2,5] considers a noisy graph signal $\boldsymbol{x} = \boldsymbol{u}+\boldsymbol{n}$ with $\boldsymbol{u}$ the signal of interest and $\boldsymbol{n}$ the noise. With the prior assumption that $\boldsymbol{u}$ varies smoothly w.r.t. the underlying graph, the denoising problem is formulated as

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}\in\mathbb{R}^N}{\operatorname{argmin}} \|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + w\,\boldsymbol{u}^\top \boldsymbol{S}\boldsymbol{u}, \tag{4}$$

with w some weighting factor between the noise suppression and smoothness prior. The optimal solution of (4) is

$$\boldsymbol{u}^* = \sum_{n=1}^{N} \left(\frac{1}{1+w\lambda_n} \boldsymbol{u}_n^\top \boldsymbol{x} \right) \boldsymbol{u}_n, \tag{5}$$

which is characterized by a rational frequency response, similar to (3), with $\psi = -w$ and $\varphi = 1$. As shown in [8], the optimal solution $\boldsymbol{u}^*$ can always be met with a *stable* implementation of (2) despite the choice of the weight[2] w.

Graph signal diffusion. Analyzing the diffusion of a graph signal, e.g., gossip in social networks, in time across the graph is another important topic in signal processing on graphs (see [2] and references therein). Such a diffusion process is often modelled using as shift operator the heat kernel [11], i.e., $e^{-w\boldsymbol{S}}$ with the constant w that characterizes the diffusion rate. Considering $\boldsymbol{x}$ as the initial signal on the graph, the diffused graph signal after t time instants is $\boldsymbol{y}_t = e^{-w\boldsymbol{S}t}\boldsymbol{x}$, with steady state ($t \rightarrow \infty$)

$$\boldsymbol{y} = (\boldsymbol{I} + w\boldsymbol{S})^{-1}\boldsymbol{x}. \tag{6}$$

We can see that the relation between the steady state diffused signal $\boldsymbol{y}$ and the initial graph signal $\boldsymbol{x}$ in (6) is also characterized by a rational polynomial in $\boldsymbol{S}$. Hence, it looks as the graph signal $\boldsymbol{x}$ is filtered by an ARMA$_1$ graph filter with frequency response (5). For processing $\boldsymbol{y}_t$ in finite time t, two dimensional graph-temporal filters can be used [12,13].

Being characterized by an ARMA$_1$ frequency response, the solutions to both Tikhonov denoising and diffusion filtering have the benefit that the filter coefficients can be designed to perfectly match the solution without relying on the knowledge of the graph structure [8]. Such a filter design is known as a *universal* design, with the main benefit that it avoids the eigendecomposition of $\boldsymbol{S}$.

Filter equivalence. As we previously mentioned, the distributed ARMA$_1$ recursion (2) will attain the frequency response (3), and thus the solution to both Tikhonov denoising and diffusion filtering theoretically at infinity. However, in practice we are interested to obtain the filter output in finite time. For $t = T$, we have

$$\boldsymbol{y}_T = (\psi\boldsymbol{S})^T\boldsymbol{y}_0 + \varphi \sum_{\tau=0}^{T-1} (\psi\boldsymbol{S})^\tau \boldsymbol{x}, \tag{7}$$

where we expanded (2) to all its terms. Depending on the choice of $\boldsymbol{y}_0$, recursion (7) has two interpretations as a FIR filter: (i) for $\boldsymbol{y}_0 = 0$, the output $\boldsymbol{y}_T$ is the same as that of an FIR$_{T-1}$ with coefficients $\boldsymbol{\phi} = [\phi_0, \phi_1, \ldots, \phi_{T-1}]^\top = [\varphi, \varphi\psi, \ldots, \varphi\psi^{T-1}]^\top$, and (ii) for $\boldsymbol{y}_0 = \boldsymbol{x}$, the output $\boldsymbol{y}_T$ is the same as that of an FIR$_T$ with coefficients $\boldsymbol{\phi} = [\varphi, \varphi\psi, \ldots, \varphi\psi^{T-1}, \psi^T]^\top$. Further, arresting the ARMA$_1$ recursion after T iterations will require the same communication and computational effort as the FIR.

3. STOCHASTIC SPARSIFICATION

This section contains our approach to perform the filtering operation in a sparsified way. The idea is that, at each iteration (time instant t), each node i will randomly choose if it will transmit the information to each of its neighbors j. In this way, the filtering costs are reduced proportionally with the probability p of selecting a neighbor. We characterize the sparsified filter output and we show that, if the filter coefficients are changed accordingly, the filter output obtained from the sparsified approach is stochastically close to the deterministic filter output obtained from the filter running on the original graph $\mathcal{G}$. Specifically, the first and second order moment of the error between them are respectively zero and upper bounded.

Stochastic graph model. Given the graph $\mathcal{G}$ that represents the the interconnections between nodes in our network of interest. We consider as a stochastic realization of $\mathcal{G}$ at time t, the graph $\mathcal{G}_t$ obtained with a random edge sampling of $\mathcal{G}$. More formally:

Random edge sampling (RES) graph model. The probability that a link (i, j) in the edge set $\mathcal{E}$ will remain active at time t is p, with $0 < p \leq 1$. The edges are activated independently accross time.

Thus, at each time step t, our graph realization $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ is a random realization of the undelying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edge set $\mathcal{E}_t \subseteq \mathcal{E}$ is generated via an i.i.d. Bernoulli process. We will indicate with $\boldsymbol{S}$ the shift operator of $\mathcal{G}$, with $\boldsymbol{S}_t$ the shift operator of $\mathcal{G}_t$ and with $\bar{\boldsymbol{S}} = \mathbb{E}[\boldsymbol{S}_t]$ the expected shift operator relative to the expected graph $\bar{\mathcal{G}}$. Since $\mathcal{E}_t \subseteq \mathcal{E}$, then also $\boldsymbol{S}_t$ belongs to the same set as $\boldsymbol{S}$. Further, due to the interlacing property [14], the spectral radius bound ϱ satisfies the property $\|\boldsymbol{S}_t\| \leq \|\boldsymbol{S}\| \leq \varrho$ for each t.

Stochastically sparsified graph filtering. The communication and computational complexity of filtering the graph signal $\boldsymbol{x}$ can be reduced by performing the filtering recursions on the time-varying sparsified graph $\mathcal{G}_t$, which abides to the RES graph model, instead of on the deterministic graph $\mathcal{G}$.

a) FIR. The sparsified output signal $\boldsymbol{y}_t^{(\mathrm{s})}$ at time instant t of an FIR$_K$ graph filter performed on the time-varying sparsfied graph is

$$\boldsymbol{y}_t^{(\mathrm{s})} = \sum_{k=0}^{K} \phi_k^{(\mathrm{s})} \boldsymbol{\Phi}_{\boldsymbol{S}}(t, t-k+1)\boldsymbol{x}, \tag{8}$$

where $\boldsymbol{\Phi}_{\boldsymbol{S}}(t', t) := \boldsymbol{S}_t\boldsymbol{S}_{t-1}\ldots\boldsymbol{S}_{t'}$ for $t' \geq t$ and $\boldsymbol{\Phi}_{\boldsymbol{S}}(t', t) := \boldsymbol{I}$ for $t' < t$ and $\phi_k^{(\mathrm{s})}$ are the sparsified filter coefficients. The FIR output in (8) is complete only for $t \geq K$ with expected output

$$\begin{aligned}\bar{\boldsymbol{y}}_t^{(\mathrm{s})} &= \mathbb{E}\left[\boldsymbol{y}_t^{(\mathrm{s})}\right] = \mathbb{E}\left[\sum_{k=0}^{K} \phi_k^{(\mathrm{s})} \left(\prod_{\tau=t}^{t-k+1} \boldsymbol{S}_\tau\right)\boldsymbol{x}\right] \\ &= \sum_{k=0}^{K} \phi_k^{(\mathrm{s})} \bar{\boldsymbol{S}}^k\boldsymbol{x} \overset{(a)}{=} \sum_{k=0}^{K} \phi_k^{(\mathrm{s})} (p\boldsymbol{S})^k \boldsymbol{x},\end{aligned} \tag{9}$$

where (a) holds for shift operators that satisfy $\mathbb{E}[\boldsymbol{S}_t] = p\boldsymbol{S}$ like $\boldsymbol{S} = \boldsymbol{A}$ or $\boldsymbol{S} = \boldsymbol{L}_{\mathrm{d}}$. Then, if each coefficient $\phi_k^{(\mathrm{s})}$ is a scaled version of ϕ_k by p^{-k}, the expected output will be identical to the FIR$_K$ output (1) performed on the original graph $\mathcal{G}$

$$\bar{\boldsymbol{y}}_t^{(\mathrm{s})} = \boldsymbol{y}_t \quad \text{for} \quad \phi_k^{(\mathrm{s})} = \phi_k p^{-k}, \tag{10}$$

[2] This result is obtained by *implementing* (5) with a shift operator that is a translated version of the normalized Laplacian, i.e., $\boldsymbol{S} = \boldsymbol{L}_{\mathrm{n}} - \boldsymbol{I}$ [8].

or equivalently, the sparsification error has a zero mean $\bar{\boldsymbol{e}} = \mathbb{E}\left[\boldsymbol{y}_t^{(\mathrm{s})} - \boldsymbol{y}_t\right] = \boldsymbol{0}$. While the appropriate modification of the filter coefficients gives a zero mean error, we next analyze the variance of the filter output to see how p influences the output signal. Our conclusions will be based on the average variance among the nodes

$$\overline{\mathrm{var}}[\boldsymbol{y}_t] = tr(\mathbb{E}\left[\boldsymbol{y}_t\boldsymbol{y}_t^{\mathsf{H}}\right] - \mathbb{E}[\boldsymbol{y}_t]\mathbb{E}[\boldsymbol{y}_t]^{\mathsf{H}})/N, \tag{11}$$

which represents a simple way to quantify the experienced variance of the filter output at each node. The following proposition (proof in Appendix), shows that the average variance of the FIR filter is upper bounded.

Proposition 1 *The average variance among the nodes of an FIR graph filter* (8) *performing the filtering recursion stocastically according to the RES graph model is upper bounded by*

$$\overline{var}[\boldsymbol{y}_t^{(s)}] \leq \left(\boldsymbol{\varrho}_p^\top \boldsymbol{\phi}\right)^2 \|\boldsymbol{x}\|^2/N, \tag{12}$$

where $\boldsymbol{\varrho}_p = [(\varrho/p)^0, (\varrho/p)^1, \ldots, (\varrho/p)^K]^\top$.

From the result of Proposition 1 we can notice that that p has indeed an impact on the variance. Further, our handle, to reduce the variance are ϱ, the spectral norm of the shift operator, and the filter coefficients $\boldsymbol{\phi}$. In this context we make the following observation:

Remark 1: To potentially reduce the variance of the output signal of the sparsified filter output, the use of shift operators with a small spectral norm is recommended. For this purpose we consider $\boldsymbol{S} = \frac{1}{\lambda_{\max}}\boldsymbol{L}_{\mathrm{d}} - 0.5\boldsymbol{I}$ (with $\varrho \leq 0.5$) for $\mathcal{G}$ and $\boldsymbol{S}_t = \frac{1}{\lambda_{\max}}\boldsymbol{L}_{\mathrm{d},t} - 0.5p\boldsymbol{I}$ for $\mathcal{G}_t$ with $\mathbb{E}[\boldsymbol{S}_t] = p\boldsymbol{S}$.

The communication and computational complexity of the sparsified FIR is now reduced linearly with p from $O(MK)$ to $O(pMK)$.

b) ARMA$_1$. Similarly to the FIR graph filter, the sparsified output for the ARMA$_1$ graph filter at time t is computed as

$$\boldsymbol{y}_t^{(\mathrm{s})} = \psi^{(\mathrm{s})}\boldsymbol{S}_t\boldsymbol{y}_{t-1}^{(\mathrm{s})} + \varphi^{(\mathrm{s})}\boldsymbol{x}, \tag{13}$$

where $\psi^{(\mathrm{s})}$ and $\varphi^{(\mathrm{s})}$ indicate the sparsified ARMA$_1$ coefficients. The expected sparsified output $\bar{\boldsymbol{y}}_t^{(\mathrm{s})}$ of the ARMA$_1$ filter at time t is

$$\bar{\boldsymbol{y}}_t^{(\mathrm{s})} = \psi^{(\mathrm{s})}(p\boldsymbol{S})\bar{\boldsymbol{y}}_{t-1}^{(\mathrm{s})} + \varphi^{(\mathrm{s})}\boldsymbol{x}, \tag{14}$$

again assuming that $\mathbb{E}[\boldsymbol{S}_t] = p\boldsymbol{S}$. In this case, if we consider $\psi^{(\mathrm{s})} = \psi/p$ and $\varphi^{(\mathrm{s})} = \varphi$ the expected output of (14) will be identical to that of the ARMA$_1$ output (2) performed on the deterministic graph $\mathcal{G}$. This change of filter coefficients will again lead to a zero mean error between the sparsified filter output (13) and the deterministic output (2).

To characterize the second order moment of the stochastic ARMA$_1$ output, we analyze the limiting average variance of the filter output over all nodes, defined as

$$\lim_{t\to\infty}\overline{\mathrm{var}}[\boldsymbol{y}_t] = \lim_{t\to\infty}\left(tr\left(\mathbb{E}\left[\boldsymbol{y}_t\boldsymbol{y}_t^{\mathsf{H}}\right] - \mathbb{E}[\boldsymbol{y}_t]\mathbb{E}[\boldsymbol{y}_t]^{\mathsf{H}}\right)/N\right), \tag{15}$$

which gives insights on the variance experienced at the steady state of the ARMA$_1$ filter. The following proposition shows that, similarly as the FIR filter, the limiting average variance among all nodes is upper bounded.

Proposition 2 *The limiting average variance among the nodes of an ARMA*$_1$ *graph filter* (13) *performing the filtering recursion stochastically according to the RES graph model is upper bounded by*

$$\lim_{t\to\infty}\overline{var}[\boldsymbol{y}_t^{(s)}] \leq \frac{1}{N}\left(\frac{\varphi^2\|\boldsymbol{x}\|^2}{(1-\varrho|\psi p^{-1}|)^2}\right). \tag{16}$$

Proof: The claim can be derived from Theorem 3 in [15], by considering that now the graph signal $\boldsymbol{x}$ is deterministic. □

Proposition 2 extends the results obtained for the FIR filters to the ARMA$_1$ graph filters. It provides a stronger result, asserting that the variance of the ARMA$_1$ output, even at infinite iterations (i.e., steady state) is upper bounded. In analogy with the filter equivalence, this can be viewed as implementing an *infinite order sparsified FIR filter* and showing that for a first order rational frequency response, the variance of the filter output is upper bounded. Differently from the FIR filters, we can see that now the limiting average variance is upper bounded by a quadratic rate of $1/p$.

Also in this case the use of the shift operator with limited spectral norm is recommended since it enlarges the stability region of the ARMA filters and thus the approximation accuracy. The sparsification approach, again reduces the distributed filtering costs of the ARMA$_1$ linearly with p, from $O(MK)$ to $O(pMK)$.

The following algorithm shows how the sparsified ARMA$_1$ filter output is computed locally at node i at time instant t.

Algorithm 1 Sparsified ARMA$_1$ filter.

```
1: Filter coefficients $\psi^{(\mathrm{s})} = \psi p^{-1}$ and $\varphi^{(\mathrm{s})} = \varphi$
2: procedure CALCULATE $[\boldsymbol{y}_t^{(\mathrm{s})}]_i$
3:     collect $[\boldsymbol{y}_{t-1}^{(\mathrm{s})}]_j$ from all current active neighbors $j \in [\mathcal{N}_i]_{t-1}$
4:     $[\boldsymbol{y}_t^{(\mathrm{s})}]_i = \psi^{(\mathrm{s})}\sum_{j\in[\mathcal{N}_i]_{t-1}} s_{ij}\left[[\boldsymbol{y}_{t-1}^{(\mathrm{s})}]_i - [\boldsymbol{y}_{t-1}^{(\mathrm{s})}]_j\right] + \varphi^{(\mathrm{s})}[\boldsymbol{x}]_i$
5: send $[\boldsymbol{y}_t^{(\mathrm{s})}]_i$ to all neighbors $\mathcal{N}_i$ with probability $p$
```

In Algorithm 1, we indicate with $\mathcal{N}_i$ the neighbouring nodes of i w.r.t. the original graph $\mathcal{G}$ (i.e., the nodes j that are connected with i by an edge with weight s_{ij}) and with $[\mathcal{N}_i]_t$ the subset of $\mathcal{N}_i$ that is connected with i at time t.

We conclude this section with the following remarks.

Remark 2: Albeit it is true that the error variance is upper bounded, numerical simulations show that the practical error variance is much smaller than the upper bounds (12) and (16). The results also confirm that p has an impact on the output variance.

Remark 3: While our mathematical results show that not using normalized shift operators, we obtain a zero mean error, the use of $\boldsymbol{S} = \boldsymbol{L}_{\mathrm{n}} - \boldsymbol{I}$ is recommended to improve the stability of the ARMA filters and also to reduce the filter output variance. However, we do not have a closed form expression of the expected value of the normalized Laplacian, i.e., $\boldsymbol{S} = \boldsymbol{L}_{\mathrm{n}}$ or its translated version $\boldsymbol{S} = \boldsymbol{L}_{\mathrm{n}} - \boldsymbol{I}$. This expression is necessary for our derivations to change the filter coefficients such that the sparsified output is not biased w.r.t. the deterministic output. Nonetheless, we have observed (and shown in the next section) that using $\boldsymbol{S} = \boldsymbol{L}_{\mathrm{n}} - \boldsymbol{I}$, *without any change in the filter coefficients*, brings a very small mean and variance of the error.

4. NUMERICAL RESULTS

In this section, we evaluate the performance of the stochastic sparsification approach. Considering that both the Tikhonov denoising problem and the graph signal diffusion can be reformulated as the same mathematical problem, we will directly address the former.

Setup. For our simulations we consider that the underlying graph $\mathcal{G}$ is composed of $N = 1000$ nodes randomly placed in a squared area, where two nodes are neighbors if they are closer than 15% of the maximum distance of the area. Our results are averaged over 100 realizations. To quantify the performance we consider the

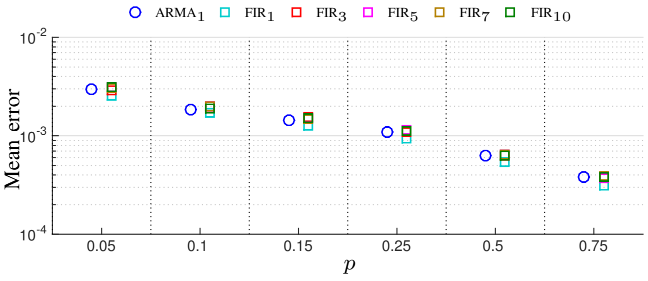

Fig. 1. Mean error over all nodes and realizations between deterministic and sparsified output for different values of p when $\boldsymbol{S} = \boldsymbol{L}_{\text{n}} - \boldsymbol{I}$.

error $\boldsymbol{e} = \boldsymbol{y}^{(s)} - \boldsymbol{y}$ between the deterministic filter output $\boldsymbol{y}$ operating on $\mathcal{G}$ and the sparsified filter output $\boldsymbol{y}^{(s)}$ operating on the RES graph realizations $\mathcal{G}_t$. We characterize both the average mean over the nodes [3] and the empirical average standard deviation over all nodes defined as

$$\overline{\text{var}}[\boldsymbol{e}] = \left[tr\left(\mathbb{E}\left[\boldsymbol{e}\boldsymbol{e}^{\mathsf{H}}\right]\right)/N\right] = \left[tr\left(\boldsymbol{\Sigma}_{\boldsymbol{y}^{(s)}}\right)/N\right], \tag{17}$$

which quantifies the average how far the filter output realization of a node is from its expected value.

Simulations. We simulate a noisy graph signal $\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{n}$, where the graph signal of interest $\boldsymbol{u}$ varies smoothly w.r.t. the underlying graph $\mathcal{G}$ characterized by the heat kernel spectrum $\hat{\boldsymbol{u}} = e^{-\lambda_n}$ with λ_n the eigenvalues of $\boldsymbol{L}_{\text{n}}$. The noise $\boldsymbol{n}$ is considered as zero mean Gaussian with standard deviation 0.1. We analyze the performance of both FIR$_K$ for $K = 1, 3, 5, 7$ and 10 and ARMA$_1$ graph filters after $t = 20 \times K$ iterations, i.e., when the transient behaviour of the filter gets close to zero. The filter operations are performed using as shift operators $\boldsymbol{S} = \frac{1}{\lambda_{\text{max}}}\boldsymbol{L}_{\text{d}} - 0.5\boldsymbol{I}$ with sparsified filter coefficients changed accordingly and $\boldsymbol{S} = \boldsymbol{L}_{\text{n}} - \boldsymbol{I}$ where the filter coefficients are not changed and kept the same as the one of the deterministic output.

Results. From the results of Fig. 1 we can see that the mean error between the sparsified filter output and the deterministic one is of order 10^{-3}, which suggests that the use of $\boldsymbol{S} = \boldsymbol{L}_{\text{n}} - \boldsymbol{I}$ does not bring a big bias in the output even without changing the filter coefficients. Fig. 2 shows the empirical average variance over nodes and realizations for both choices of the shift operator. We can make the following observations: (i) the ARMA$_1$ graph filter seems to survive better the stochastic sparsification than FIR for both choices of $\boldsymbol{S}$; (ii) for the FIR filter the choice of $\boldsymbol{S}$ has a bigger impact on the variance, where $\boldsymbol{S} = \frac{1}{\lambda_{\text{max}}}\boldsymbol{L}_{\text{d}} - 0.5\boldsymbol{I}$, even with the guarantees that the bias is zero comes with a higher variance; (iii) with the order increasing, the FIR filter output experiences a higher variance.

For the ARMA$_1$, the bound (15) for the considered values of p is $[0.0121, 0.1477, 5.3127, 0.5908, 0.1477, 0.1085]$. On the other hand, for the FIR filter the bound (12) tends to be loose for $p < 0.25$ and for $K > 7$. For the choice of $p = 0.25$ and for the considered values of K the bound (12) is $[0.0014, 0.0113, 0.0532, 0.2267, 1.4570]$.

To wrap up, recalling also the filtering equivalence, the use of the ARMA$_1$ filter *for the aforementioned tasks* can replace any FIR filter and it can be implemented distributively in a stochastic sparsification fashion saving up to 95% of communication and computational costs with a bias and variance of order 10^{-3}. As future research we aim evaluating influence that the graph topology has on the proposed framework.

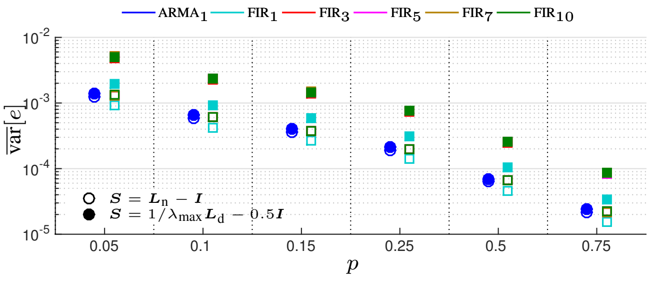

Fig. 2. Average empirical variance among all nodes of the error between the sparsified filter output and the original graph output for different values of p.

5. CONCLUSIONS

In this work we have presented a novel approach on how to sparsify the graph filtering operation in a stochastic way. Since the filter output will now be stochastic, we show that if the filter coefficients are changed accordingly the expected output is identical to the one of the deterministic filter operating in a non-sparsified way. Further, we show that the variance of the sparsified output is upper bounded. Numerical results show that for the particular tasks of graph signal denoising and diffusion the distributed costs can be reduced up to 95% differing very little from the original output.

6. APPENDIX

Proof of Proposition 1 We start by computing the trace of the covariance matrix of the filter output at time t as

$$tr(\boldsymbol{\Sigma}_{\boldsymbol{y}}[t]) = tr(\mathbb{E}[\boldsymbol{y}_t^{(s)}\boldsymbol{y}_t^{(s)\mathsf{H}}]) - tr(\mathbb{E}[\boldsymbol{y}_t^{(s)}]\mathbb{E}[\boldsymbol{y}_t^{(s)}]^{\mathsf{H}}). \tag{18}$$

By using the linearity of the expectation and the trace and substituting the expression of $\boldsymbol{y}_t^{(s)}$ (8), we can expand the first term on the right hand side of (18) as

$$tr(\mathbb{E}[\boldsymbol{y}_t^{(s)}\boldsymbol{y}_t^{(s)\mathsf{H}}]) = \sum_{k=0,\kappa=0}^{K} \phi_k^{(s)}\phi_\kappa^{(s)} T_{k,\kappa}(\boldsymbol{x}, \boldsymbol{S}), \tag{19}$$

with

$$T_{k,\kappa}(\boldsymbol{x}, \boldsymbol{S}) = tr\left(\boldsymbol{x}\boldsymbol{x}^{\mathsf{H}}\mathbb{E}\left[\boldsymbol{\Phi}_{\boldsymbol{S}}(t, t-\kappa+1)^{\mathsf{H}}\boldsymbol{\Phi}_{\boldsymbol{S}}(t, t-k+1)\right]\right), \tag{20}$$

where in (20) we have commuted the trace and the expectation and also have applied the circular property[4] of the trace. We can therefore use the inequality

$$tr\left(\boldsymbol{A}\boldsymbol{B}\right) \leq 0.5\|\boldsymbol{A} + \boldsymbol{A}^{\mathsf{H}}\|tr(\boldsymbol{B}) \leq \|\boldsymbol{A}\|tr(\boldsymbol{B}) \tag{21}$$

valid for any symmetric matrix $\boldsymbol{A}$ and positive semi-definite matrix $\boldsymbol{B}$ of appropriate dimensions [16], together with the triangle inequality of the norms and the fact that the realizations of the shift operator are upper bounded as $\|\boldsymbol{S}_t\| \leq \varrho$. Then, we can bound $T_{k,\kappa}(\boldsymbol{x}, \boldsymbol{S})$ as

$$T_{k,\kappa}(\boldsymbol{x}, \boldsymbol{S}) \leq tr\left(\boldsymbol{x}\boldsymbol{x}^{\mathsf{H}}\right)\|\mathbb{E}\left[\left(\prod_{\tau=t}^{t-\kappa+1}\boldsymbol{S}_\tau\right)\left(\prod_{\tau=t}^{t-k+1}\boldsymbol{S}_\tau\right)\right]\| \leq \|\boldsymbol{x}\|^2\varrho^{\kappa+k}. \tag{22}$$

The second term in the right hand side of (18), is positive so it can be lower bounded by 0. Then, substituting the latter, (22) and (19) into (18) and with simple algebra the upper bound (12) follows. □

[3] Note that for $\boldsymbol{S} = \frac{1}{\lambda_{\text{max}}}\boldsymbol{L}_{\text{d}} - 0.5\boldsymbol{I}$ and changing the filter coefficients according to Section 3 this is 0.

[4] $tr(\boldsymbol{ABC}) = tr(\boldsymbol{CAB}) = tr(\boldsymbol{BCA})$.

7. REFERENCES

[1] M. E. J. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[3] A. Sandryhaila and J. M. Moura, "Discrete Signal Processing on Graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[4] ——, "Discrete Signal Processing on Graphs: Frequency Analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[5] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovacevic, "Signal Denoising on Graphs via Graph Filtering," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, USA, December 2014, pp. 872–876.

[6] J. Ma, W. Huang, S. Segarra, and A. Ribeiro, "Diffusion Filtering of Graph Signals and Its Use in Recommendation Systems," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4563–4567.

[7] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed Linear Network Operators Using Graph Filters," *arXiv preprint arXiv:1510.03947*, 2015.

[8] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive Moving Average Graph Filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2017.

[9] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized Gossip Algorithms," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2508–2530, 2006.

[10] D. A. Spielman and S.-H. Teng, "Spectral Sparsification of Graphs," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 981–1025, 2011.

[11] A. J. Smola and R. Kondor, "Kernels and Regularization on Graphs," in *Learning theory and kernel machines*. Springer, 2003, pp. 144–158.

[12] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable Autoregressive Moving Average Graph-Temporal Filters," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 200–204.

[13] E. Isufi, G. Leus, and P. Banelli, "2-Dimensional Finite Impulse Response Graph-Temporal Filters," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washington D.C, USA, 2016.

[14] G. Chen, G. Davis, F. Hall, Z. Li, K. Patel, and M. Stewart, "An Interlacing Result on Normalized Laplacians," *SIAM Journal on Discrete Mathematics*, vol. 18, no. 2, pp. 353–361, 2004.

[15] E. Isufi, A. Simonetto, A. Loukas, and G. Leus, "Stochastic Graph Filtering on Time-Varying Graphs," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015.

[16] S.-D. Wang, T.-S. Kuo, and C.-F. Hsu, "Trace Bounds on the Solution of the Algebraic Matrix Riccati and Lyapunov Equation," *Transactions on Automatic Control*, vol. AC-31, no. 7, pp. 654–656, 1986.